#### UNIVERSITY OF NEVADA LAS VEGAS. DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING LABORATORIES.

Class:	СР	E300L	Semester:	Fall 2021							
					-						
Points		Document author:	Jerrod Batu								
		Author's email:	batuj1@unlv.nevada.edu								
			-								
		Document topic:	Postlab 9								
Instructor's	s com	iments:									

# 1. Introduction / Theory of Operation

In this lab, I will become familiar with the design of a complete CPU by implementing and experimenting with single cycle implementation of MIPS instructions.

- a. The **main elements** of a single cycle datapath include the control unit, datapath, data memory, and instruction memory. The control unit will decode the instructions and provide control signals for the datapath. The datapath will receive the control signals from the control unit to write data and calculate address into the data memory or to provide PC (program counter) address to instruction memory. The data memory receives the address and data from the datapath and outputs the data to be decoded from the control unit. The instruction memory outputs instruction data to be read into the datapath.
- b. The control unit circuit which generates the control signals is **combinational**. The control unit circuit is combinational because it combines two decoders: main decoder and alu decoder. This control unit will also produce the AND gate control signal for the program counter source. These decoders will determine which instruction to be performed given by the instruction memory. Another reason why this control unit circuit is combinational is because it is does not have a clock, reset, or start input; therefore, it is not an FSM that is dependent on the clock cycle, but it is rather a set of combinational units to determine which instruction to be performed whether it be add, sub, addi, and, or, slt, beq, or j instructions.
- c. There are a few **other functional units that are to be added to the ALU.** Since the ALUop is decoded with two bits, 00 for add and 01 for sub, 10 would perform these other functional units to the ALU. These instructions include add, addu, and, nor, or, sub, subu, xor, slt, sltu, lb, lbu, lh, lhu, lw, sb, sh, or sw. Primarily, the ALU is capable of storing and loading words into the register file. In addition, the ALU can also perform gate logic operations such as and, nor, or, and xor.

#### 2. Prelab

https://docs.google.com/document/d/1W3bFRnSusdZvWCfRua07DCCUL3C\_EYok/edit?usp=sh aring&ouid=102808507017671072128&rtpof=true&sd=true

This is the link to my prelab 9 document.

# 3. Results of Experiments

#### EXPERIMENT 1

```
module testbench();
reg clk;
reg reset;
wire [31:0] writedata, dataadr;
wire memwrite;
// instantiates the device to be tested (device under test-DUT)
MIPSsubset dut (clk, reset, writedata, dataadr, memwrite);
// initialize test
initial
begin
reset <= 1; # 22; reset <= 0;
end
// generate clock to sequence tests
always
begin
clk \ll 1;
# 5;
clk <= 0;
# 5; // clock duration
end
// check the contents of memory location 84. It must be 7
always @ (negedge clk)
begin
if (memwrite) begin
if (dataadr === 84 & writedata === 7) begin
$display ("Simulation succeeded");
$stop;
end
else if (dataadr !== 80) begin
$display ("Simulation failed");
$stop;
end
end
end
endmodule
```

This is my testbench module for the MIPS subset.

,		 ▼♯Σ驥Σ	C1:0 REF															Sir (1	n:175 75)
Name	e	Value	0			40		60 	<sup>18</sup>			100		120	Lini	140		160	
÷ G	iroup1																		
	- O cik	St1																	
	- 🛙 reset	St1																	
	memwrite	St0																	
ŧ	<u>∔</u>	32'h0000_0005	0000_0005	00	000_000c	0000_0003	0000_0007	0000_0004 <mark>)</mark> (	0000_000b	0000_0008	0000_	0000	0000_0001	0000_0000	0000_0007	0000	_0050	0000_0000	*54
ŧ	≟n_writedata[31:0]	32'hxxxx_xxxx	0000_	005			0000_0005	0000_0	0007	0000_0003	0000_0007	0000_0000	0000_0005	0000_0001	0000_0005	0000_0007	0000_0005	0000_0000	×07
Ni	Jew Group																		

These are my VCS waveforms for the MIPS subset.



# EXPERIMENT 2

This is a picture of my DE2 board displaying the content of \$2 (7) being stored into the memory location (84) at the corresponding address (44).

<u>https://drive.google.com/file/d/1U0fXFHva\_ePXsBLyd4B7FiNW13MMI7Pz/view?usp=sharing</u> This is the link to my video explaining each MIPS subset instruction performed on the DE2 board.

# EXPERIMENT 3

File View Netlist Constraints Reports Scrip	ot Tools Wind	low Help 🐬	9										Search altera.com
Report # #	Setup: dk												(
TimeQuest Timing Analyzer Summary	Command Inf	o Summary o	of Paths										
ar Seup. uk	Slack		From Node			To Node		Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay	ay
	1 -7.398	dmem:dmem F	RAM~77		mips:mips data	path:dp regfile:r	rf rf~109	dk	dk	1.000	0.004	8.438	
	2 -7.269	dmem:dmem  F	RAM~77		mips:mips data	path:dp regfile:r	f rf~45	dk	dk	1.000	0.004	8.309	
	3 -7.248	mips:mips[data	apath:dp[flopr:p	pcreg[q[3]	dmem:dmem R/	AM~1156		dk	dk	1.000	-0.331	7.953	
	4 -7.242	mips:mips data	apath:dp flopr:p	pcreg[q[3]	dmem:dmem R	AM~1252		dk	dk	1.000	-0.331	7.947	
	5 -7.229	amem:amemji	CAM~77		mips:mips(data)	path:dpjregfile:r	T[[T~//	OK	CK	1.000	-0.007	8.258	
	7 -7 204	dmem:dmem/F	CAM~77		mips:mips/data	path:dplregfile:r	flrfw141	dk	dk	1.000	-0.025	8 215	
Tasks 무 루 ×	8 -7.199	dmem:dmem  F	RAM~77		mips:mips data	path:dplregfile:r	firf~205	dk	dk	1.000	-0.025	8.210	
🖋 🙀 Open Project 🔥	9 -7.199	dmem:dmem  F	RAM~77		mips:mips data	path:dp regfile:r	f rf~173	dk	dk	1.000	-0.025	8.210	
Detist Setup	10 -7.191	mips:mips data	apath:dp flopr:p	ocreg q[3]	dmem:dmem R	AM~243		dk	dk	1.000	-0.339	7.888	
🗸 🕨 Create Timing Netlist													
Read SDC File	Path #1. Set	un elack is -73									Path #	1. Setun slack	-k is +7 398 (VTOLATED)
Update Timing Netlist	Path Summar	v Statistics	Data Path	Wavefor	0						Path	Summary Sta	tatictics Data Path Waveform
Set Operating Conditions		, Deserve	Value	Count	Tatal Dalau	9/ of Tabal	Min	Mau				ourinda y ota	
Reports	1 500	in Delationship	1 000	Count	Total Delay	76 01 10tal	Part	Max					
Y De Slack	2 Clor	k Skew	0.004										2.113 ns
Report Setup Summary	3 Dat	a Delay	8.438									L as much	
Report Hold Summary	4 Nun	ber of Logic Leve	ds	7							Launci	h Clock Calance	
Report Recovery Summary	5 Y Phy	sical Delays											1.0 ms
Report Removal Summary	1 ~	Arrival Path									Setup	Relationship	p
📷 Report Minimum Pulse Width Summa	1	Y Clock		2		10	0.440	1.000					
Report Max Skew Summary	1	IC Coll		2	1.157	42	0.118	1.039			Latch	Clock	Laton
Datasheet	2	Y Data		5	1.550	37	0.000	0.555					
Report Fmax Summary	1	IC		8	6,484	76	0.247	1.624			Data	Arrival	
Report Datasheet	2	Cel		9	1.704	20	0.000	0.438					
Device specific	3	uTco		1	0.250	2	0.250	0.250			Clock	Delay	2,693 ns
Report ICCS	2 ~	Required Path											
Report NDR	1	✓ Clock		_							Data	Delay	8.438 ns
Report Metastability	1	IC		2	1.161	43	0.118	1.043					
Diagnostic	2	Cel		3	1.536	50	0.000	0.999			Slack		-7.398 ns
Report Clocks													
Report Clock Transfers											Data	Required	Y
Report Unconstrained Paths													·
Report SDC											Clock	Delay	2,697 ns
Report Ignored Constraints													
Check Timing											uTsu		-0.036 hs
Report Partitions	Note: Negative	e delays are omit	ted from totals v	when calcula	ting percentages						- arou		
<ul> <li>Custom Reports</li> </ul>											Tine	(ns)	0.0 1.0 2.4 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11
Report Timing													
< >													

This is the TimeQuest analyzer for the MIPS subset before setting the timing constraints.

e View Netlist Constraints Reports Script	Tools Window He	ielp 💎															Search altera.co	um 🧃	
port 48	Setup: dk																	0	
TimeQuest Timing Analyzer Summary	Command Info Su	ummary of Paths																	
Seup: uk	Slack From Node				To Node		Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay								
	1 0.602 dmem	n:dmem IRAM~77		mips:mipsIdata	path:dplreofile:r	firf~109	dk	dk	9.000	0.004	8,438								
	2 0.731 dmem	n:dmemIRAM~77		mips:mipsIdata	path:dplreofile:r	firf~45	dk	dk	9.000	0.004	8.309								
	3 0.752 mips:	mips datapath:dp fi	opr:pcreg[g[3]	dmem:dmem  R	AM~1156		dk	dk	9.000	-0.331	7.953								
	4 0.758 mips:	mips datapath:dp fi	opr:pcreg[g[3]	dmem:dmem]R	AM~1252		dk	dk	9.000	-0.331	7.947								
	5 0.771 dmem	n:dmem RAM~77		mips:mips data	path:dp]regfile:r	frf~77	dk	dk	9.000	-0.007	8.258								
	6 0.796 dmem:dmem RAM~77			mips:mips data	path:dp regfile:r	frf~13	dk	dk	9.000	-0.025	8.215								
	7 0.796 dmem	n:dmem RAM~77		mips:mips data	path:dp regfile:r	f rf~141	dk	dk	9.000	-0.025	8.215								
sks 부분×	8 0.801 dmem	n:dmem RAM~77		mips:mips data	path:dp/regfile:r	frf~205	dk	dk	9.000	-0.025	8.210								
🞯 Open Project 🔨	9 0.801 dmem	n:dmem RAM~77		mips:mips data	path:dp regfile:r	f rf~173	dk	dk	9.000	-0.025	8.210								
Netlist Setup	10 0.809 mips:	mips datapath:dp fi	opr:pcreg[q[3]	dmem:dmem R	AM~243		dk	dk	9.000	-0.339	7.888								
Create Timing Netlist																			
Read SDC File																			
Update Timing Netlist	Update Timing Netlist Path #1: Setup slack is 0.602									Path #	1: Setup slack is	s 0.602							
Reset Design	Path Summary St	tatistics Data Pat	th Wavefo	rm						Path S	ummary Stati:	stics Data Path Wavef	orm						
Set Operating Conditions	Prope	erty Val	lue Count	Total Delay	% of Total	Min	Max												
C Reports	1 Setup Relation	ionship 9.00	0																
Y 🦢 Slack	2 Clock Skew							-2.146 n	IS .										
Report Setup Summary	3 Data Delay	8.43	8							a sum als	Launch					1			
Report Hold Summary	4 Number of Le	ogic Levels	7							Launch	I LIOCK LOUAITOIT								
📷 Report Recovery Summary	5 Y Physical Dela 5	ays												9.0 pc					
Report Removal Summary	1 Y Arrival P	Path								Setup	Relationship			210 112					
式 Report Minimum Pulse Width Summa	1 Y Cloc	dk																	
Report Max Skew Summary	1	IC	2	1.157	42	0.118	1.039			Latch	Clock					Latch			
Y 📴 Datasheet	2	Cell	3	1.536	57	0.000	0.999				-								
Report Fmax Summary	2 Y Data	а								Data A									
Report Datasheet	1	IC	8	6.484	76	0.247	1.624			oucu n									
Device Specific	2	Cel	9	1.704	20	0.000	0.438					2.693 ns							
Report TCCS	3	uTco	1	0.250	2	0.250	0.250			Clock	uelay		*						
Report RSKM	2 Y Required	d Path													0.420				
Report DDR	1	ok .			40	0.110	1.0.0			Data D	lelay				8.438 ns				
Report Metastability	1	IC C	2	1.161	43	0.118	1.043												
Y Pianostic	2	Cel	3	1.536	30	0.000	0.333			Slack								0.602 r	
Report Clorke																			
Report Clock Transferr										Date D	-								
Depart Upcentrained Path-										uata R	equired								
Depart SDC																	2 697 pt		
Depart Jacob Combaint										Clock	Delay							í	
Report Ignored Constraints																			
Et Cneck Timing										uTsu								-0.03	
BU Report Partitions	Note: Negative delays	are omitted from to	tals when calcu	lating percentages															
<ul> <li>Custom Reports</li> </ul>										Tine (	ins) 0.	.0 1.125 2.25	3.375	4.5 5	5.625 6.75	7.875 9.	.0 10.125	11.25	
Descet Teslan																			

This is the TimeQuest analyzer for the MIPS subset after setting the timing constraint of 9 ns.

Cycle time: 9 ns

# of instructions: 15 instructions

Total time to complete code: (9ns) x (15 instructions) = 135ns

#### 4. Answers to questions

Question 1:

Compared to other processor architectures, MIPS is very simple to understand and use. Since MIPS is RISC, there are a reduced instruction set of computers that are more optimized by the compilers. Compared to DEL, CISC, and VLIW, MIPS has the least amount of instruction sets. In addition, there is a flexibility of high-performance cache and memory management that allows MIPS to have a strong advantage over any other processor architectures. MIPS is also distinguished from its large number of registers, uniform instruction length, and fewer instruction formats.

#### Question 2:

Advantages -

- 1. MIPS delivers lower power consumption than other processor architectures.
- 2. MIPS architecture offers flexibility of high-performance caches and memory management.

3. MIPS is one of the simplest architectures to introduce computing architecture. Disadvantages -

- 1. Each operation would require memory access; therefore, the processor needs to be slow or the memory must be small.
- 2. Instructions must encode memory addresses rather than register numbers, so the size of each instruction would need to be larger to access all memory addresses.
- 3. MIPS requires a mode switch before any 16-bit instructions can be executed.

# 5. Conclusions & Summary

Overall, this lab was very similar to what was being taught in the lecture. I am still trying to grasp the concept of the MIPS implementation in Verilog through several different components. I would say that the Verilog module instantiations are somewhat of a more complex general datapath. It is very interesting to learn how one architecture language can be implemented into a high coding language. There were not any issues or problems I had when performing the experiments for the lab, and I hope to learn more about how MIPS instructions can be incorporated into Verilog modules.